



## Connectivity management in mobile ad hoc networks using particle swarm optimization

Orhan Dengiz<sup>a</sup>, Abdullah Konak<sup>b,\*</sup>, Alice E. Smith<sup>c,1</sup>

<sup>a</sup> DnD Technical Solutions, Tunus Cad. 87-5, Kavaklıdere Ankara 06680, Turkey

<sup>b</sup> Information Sciences and Technology, Penn State Berks, Tulpehocken Rd., PO Box 7009, Reading, PA 19610, United States

<sup>c</sup> Department of Industrial and Systems Engineering, 3301 Shelby Center, Auburn University, AL 36849-5346, United States

### ARTICLE INFO

#### Article history:

Received 18 February 2010

Received in revised form 29 December 2010

Accepted 19 January 2011

Available online 10 March 2011

#### Keywords:

Ad hoc networks

Network connectivity

Particle swarm optimization

Location models

### ABSTRACT

This paper proposes a dynamic mobile ad hoc network (MANET) management system to improve network connectivity by using controlled network nodes, called agents. Agents have predefined wireless communication capabilities similar to the other nodes in the MANET, however their movements, and thus their locations, are dynamically determined to optimize network connectivity. A new approach to measuring connectivity using a maximum flow formulation is proposed – this is both responsive and tractable. Furthermore, users' locations are predicted for several time steps ahead and this is shown to improve network connectivity over the network operation period. A particle swarm optimization (PSO) algorithm uses the maximum flow objective to choose optimal locations of the agents during each time step of network operation. The proposed MANET management system is rigorously tested on numerous static and dynamic problems. Computational results show that the proposed approach is effective in improving the connectivity of MANETs and predicting movements of user nodes and deploying agents accordingly significantly improves the overall performance of a MANET.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Mobile ad hoc networks (MANET) are formed without a central administration so that nodes transmit packets on behalf of other nodes. The most commonly cited application of MANET is military communications including combat, emergency response, and search/rescue maneuvers [1–5].

There are many factors that affect the performance and reliability of a MANET. Links between the mobile devices sometimes exist, and sometimes do not, depending on the devices' locations relative to each other, their transmission power and the surrounding environment. New mobile

devices can enter the system or existing devices can leave for various reasons including loss of battery power or loss of signal strength (due to distance or environmental causes). Assuming random user behavior, it is very likely that one or more users will lose connectivity with the network or with the parts of the network due to their positions relative to other users. If a user is outside the range of its nearest neighbor in terms of signal strength, then its access to the rest of the network is unavailable.

This paper proposes a dynamic MANET management system to maintain the connectivity of a MANET by using controlled ad hoc network nodes (called agents), where the global state of the network can be tracked using a Global Positioning System (GPS). In a GPS, each node is equipped with a receiver and the location data is periodically transmitted to a central location using low frequency radio or a satellite modem embedded in the node [6]. In this paper, user nodes are allowed to move freely and their

\* Corresponding author. Tel.: +1 610 396 6310; fax: +1 610 396 6024.

E-mail addresses: [odengiz@dndltd.com.tr](mailto:odengiz@dndltd.com.tr) (O. Dengiz), [konak@psu.edu](mailto:konak@psu.edu) (A. Konak), [smithae@auburn.edu](mailto:smithae@auburn.edu) (A.E. Smith).

<sup>1</sup> Tel.: +1 334 844 1400; fax: +1 334 844 1381.

## Nomenclature

$UN_t$	the set of user nodes at time $t$	$e_{(i,j,t)}$	1 if there exists a link between nodes $i$ and $j$ at time $t$ , 0 otherwise
$NU$	the number of user nodes	$u_{ijt}$	the capacity (bps) of link $(i, j)$ at time $t$
$NU_t$	the number of user nodes at time $t$	$(x_{it}, y_{it})$	the $x$ and $y$ coordinates of the $i$ th MANET node at time $t$
$AN_t$	the set of mobile agent nodes at time $t$	$(x_{it}^*, y_{it}^*)$	the $x$ and $y$ coordinates of the $i$ th agent node for the best solution at time $t$
$NA$	the number of agent nodes	$XY_t$	the set of $x$ and $y$ coordinates of the MANET nodes at time $t$
$NA_t$	the number of agents nodes at time $t$	$v_{\max}$	the maximum Euclidean distance units traveled per unit time increment (note that any distance metric could be used)
$N_t$	the set of all MANET nodes at time $t$ ( $N_t = UN_t \cup AN_t$ )	$H$	prediction horizon (in unit time) for the location of user nodes
$G_t$	the network at time $t$	$U(a, b)$	uniform random variable between $a$ and $b$
$NC$	the number of user node clusters		
$E_t$	the set of links between all MANET nodes at time $t$		
$t_0$	initial time		
$R_{it}$	the wireless connection range of the $i$ th node at time $t$		

current and past location data are available by the use of the GPS and a kinematic-based prediction model used to estimate their future locations. The purpose of agents is to augment network connectivity. Agents are moved each time step during the operation of the network to optimize network connectivity among user nodes. (Note that networks operate in continuous time, however to make the optimization tractable the continuous time operation is sliced into discrete time steps.) The predicted locations of users are used in a particle swarm optimization (PSO) algorithm to dynamically direct the motion of the mobile agents. Among application areas of the proposed approach are network-centric warfare military, fleet tracking, and search and rescue operations, where network nodes have power sources to support a GPS.

There has been very limited work in the literature to improve network connectivity through mobile agents. In this paper, multiple mobile agents are considered and no restrictions are imposed on the movement of user nodes. The proposed approach is applicable to all user movement models and scenarios. The objective function is a network connectivity measure using a maximum flow formulation that aims to increase overall network performance, independent from the routing protocol used. Therefore, the problem is studied for a much more general case than previously published work.

## 2. Background

In the literature, several approaches have been proposed to address the challenges in MANETs due to unpredictable user node movements. One of the major problems is the accessibility of the centralized network services used by all network nodes when the network is disconnected. A solution approach to this problem involves replicating network services [7–8] and critical data [9] at multiple nodes and dynamically deploying these nodes to disconnected partitions of the network. Another problem is delivery of data packets across disconnected network partitions. To address this problem, a few papers [10–11]

propose using special agent nodes that can buffer packets until their destination nodes are reachable. These agent nodes may move arbitrarily or systematically, and when they are connected to a network partition, they deliver their payloads. This approach, however, is mainly applicable to delay-tolerant data networks. Alternatively, several papers propose topology control by modifying node trajectories or power level nodes. In [12], the intermediate nodes between a source and a destination node modify their trajectories to ensure the delivery of packets, and an algorithm to minimize the trajectory modifications is proposed. Goyal and Caffery [13] propose first determining critical links whose failures cause partitioning of the network, and then supporting these links by either modifying the trajectory of the nodes involved in the critical links or bringing an outside node to reinforce them. Kim et al. [14] use transmission power management schemes to strengthen the network topology across the critical nodes of a network. Montemanni and Gambardella [15] also propose a distributed algorithm to assigning transmission powers in wireless networks in such a way that all the nodes are connected and the total power consumption of the network is minimized.

There has been limited work in the literature to improve network connectivity in MANETs through mobile agents. Ou et al. [16] uses forwarding nodes that can adjust their locations to assist network partitions in an ad hoc network. When deployed initially, forwarding nodes move randomly to discover network partitions. It is assumed that each node has GPS capability and the nodes can exchange location data. When a forwarding node receives a service request, it will move to the service area if it is available (i.e., it is not servicing other nodes). However, the paths of the forwarding nodes are not optimized in this approach. Chadrashekar et al. [17] define the problem of achieving connectivity of a disconnected ground MANET by dynamically placing unmanned air vehicles (UAVs) which function as relay nodes. Then, they develop a heuristic algorithm for the single-UAV problem. Zhu et al. [18] also propose using UAVs equipped with communication capabilities to provide services to ground-based MANETs.

They define a gradient-based algorithm to determine the location of a single-UAV to maximize two network connectivity measures, global message connectivity and worst-case connectivity. Both connectivity measures are defined on a spanning tree so the optimization problem is to determine the most reliable Steiner tree of the network, where the UAV is considered a Steiner point in the network. Hauert et al. [19] propose the deployment of a swarm of UAVs for search and rescue missions. During a mission, UAVs are expected to maintain direct or indirect connection to their base-station through the ad hoc network that they form. An ACO approach is developed to control the UAV MANET. In this paper, there are no user nodes that move – the network is made of only agents and a fixed target.

The proposed approach herein this paper is also related with the work on self-organizing particle systems. In these systems, the objective is to model the collective movements of locally interacting particles such as bird flocks and insect swarms using computational models. A group-flocking behavior can be created from simple interactions of autonomous particles. For example, Reynolds [20] defines three simple rules, “attempt to stay close to nearby flockmates,” “avoid collisions with nearby flockmates,” “attempt to match velocity with nearby flockmates” to achieve simulated flocking. Recently, self-organizing particle systems have been applied to several problems such as multi-robot teams [21–22], target tracking in sensor networks [23], and target search [24]. The focus of these approaches is mainly on coordinating a collective movement of the swarm while avoiding obstacles and ensuring that particles do not collide. There has been limited work on autonomous problem solving using self-organizing particles. For example, Moarref and Sayyadi [25] propose an asynchronous, multi-agent robotic system approach to the continuous  $n$ -median problem, which is defined as locating  $n$  facilities on a continuous plane so that the weighted sum of the Euclidean distances of a set of target points to their nearest facility is minimized. In their approach, each facility moves toward the geometric median of its corresponding Voronoi cell as the target points change locations. Rodriguez and Reggia [26] use multiple agents to solve resource locate-and-collect problems. In their approach, agents have the capability to solve small problems in addition to flocking behavior.

### 3. Problem formulation

#### 3.1. Assumptions and problem inputs

There are two main types of MANET nodes; user nodes and agent nodes. User nodes are the nodes that demand network service. Mobile agents are responsible for helping the user nodes experience the best network service possible. The user nodes in the MANET move at their own will and it is assumed that their future positions are unknown. However, the location data of users and agents are assumed to be available to the agent control system at all times. This is technically possible by using a GPS [6]. Every node has a certain wireless connection range and a maximum velocity.

For any time  $t$  and set  $XY_t = \{(x_{it}, y_{it}) : i \in N_t\}$ , the network  $G_t = (N_t, E_t)$  is formed as follows:

$$e_{(i,j,t)} = \begin{cases} 1 & \text{if } R_{it} \geq d_{ijt} \text{ and } R_{jt} \geq d_{ijt} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $i, j \in N_t$  and  $d_{ijt}$  is the Euclidean distance between nodes  $i$  and  $j$  at time  $t$  as follows:

$$d_{ijt} = \sqrt{(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2} \quad (2)$$

In a wireless network, the capacity of a link depends on the signal strength, which is a function of the link distance and some external factors. In general, the link distance and signal strength, and thus the link data rate, are negatively correlated. The wireless IEEE 802.11 standard is capable of linking MANET nodes [1]. Using this protocol, it is technically feasible to create a multi-hop network that covers several square kilometers [2]. The path loss model [27] can be used along with the product specifications of wireless equipment vendors to estimate link data rates for given distances. Without loss of generality, normalized transmission ranges and data rates are used in this paper based on an industry leader’s data [28]. The relationship between the distance and link capacities in terms of data rate in bits per second (bps) is approximated with a continuous function as follows:

$$u_{ijt} = (1 + e^{10 \cdot (d_{ijt} - 0.5)})^{-1} \quad (3)$$

This capacity function conforms to the basic requirements of the normalized distance versus the data rate, that is, when the distance is close to zero, the normalized data rate is close to one, and when the distance is close to the wireless transmission range, the data rate is close to zero. An exponential decrease of the data rate occurs as the distance increases, as is observed in practice. The function given in (3) may not be the most accurate estimation of the normalized data rates at intermediate distances, as the path loss and the data rate estimation models are estimates assuming constant interference and certain environmental conditions. In addition, technical capabilities such as antenna reception of devices differ and the specification curve defined in (3) will shift left or right for different products. Note that the proposed approach is independent of the path loss function and equipment. More complex path loss functions, e.g. considering obstacles and antenna gains, could be used instead of (3). Similarly a different data rate model could be adapted to estimate link capacities.

#### 3.2. Evaluating network connectivity

The overall objective is to increase the connectivity of the MANET. In the ideal case, there should be at least one path between every user node pair in the network. Therefore, the primary objective is to connect each user node to every other one. This overall connectivity objective can be expressed as

$$O_{1t} = \frac{2 \sum_{i,j \in UN_t; j>i} z_{ijt}}{NU \times (NU - 1)} \quad (4)$$

where  $z_{ijt} = 1$  if there is a path between the  $i$ th and  $j$ th user nodes at time  $t$ , and  $z_{ijt} = 0$ , otherwise.

Note that (4) yields the same value for all networks with the same number of users connected. Therefore, it is not useful for distinguishing among connected networks. A good measure to quantify the connectivity between two nodes  $S$  and  $T$  is the cardinality of their minimum cut set. This measure gives the minimum number of links that needs to be deleted to render nodes  $S$  and  $T$  disconnected. A drawback of the cardinality of the minimum cut set as a connectivity measure is that it does not quantify the quality of the links, which is an important concern in wireless communication. In this paper, a MANET at any time  $t$  is modeled as a capacitated network with link data rates (capacities) which are the indicators of link qualities. Let  $U(G_t, S, T)$  be the maximum flow (in bps) that can be sent from node  $S$  to node  $T$  through all possible paths between these two nodes on network  $G_t$ .  $U(G_t, S, T)$  provides a good measure of the connectivity between nodes  $S$  and  $T$ , as well as the quality of the links between them.  $U(G_t, S, T)$  can be calculated by solving a maximum flow problem with  $S$  as the source node and  $T$  as the sink node on a capacitated network  $G_t$ . The minimum of those maximum flows between every user pair as expressed in Eq. (5) can be used to gauge network connectivity.

$$O_{2t} = \min_{i,j \in UN_t; i > j} \{U(G_t, i, j) : U(G_t, i, j) > 0\} \quad (5)$$

Maximizing  $O_{2t}$  improves the weakest connections in the network. The maximum flow problem is a well-known network optimization problem, and there are various algorithms readily available to optimally solve it, including ones in polynomial time [29,30]. However, solving the maximum flow problem to calculate  $O_{2t}$  is a computationally expensive operation, particularly for large problems considering all node pairs. Since the network operates dynamically in real time, a good solution must be quickly found during each time step so that agents can be properly deployed to their new locations. To reduce the computational time to calculate  $O_{2t}$ , a surrogate objective function is used as follows. First,  $NC$  clusters of the user nodes are determined using a clustering algorithm based on the minimum spanning tree of the user nodes, where the Euclidean distances between the user nodes are used as link costs. As described in Ahuja et al. [30],  $NC$  disconnected clusters of the user nodes are obtained by deleting  $(NC-1)$  links of the minimum spanning tree one by one in decreasing order of their distances. Then, the maximum flows among cluster pairs are computed and the minimum of them is used as the surrogate objective function for  $O_{2t}$ . An example for user node clusters and cut sets are illustrated in Fig. 1. In the computational experiments section, it is shown that this approach provides a very good approximation to (5) in dynamic problem scenarios.

Maximizing  $O_{2t}$  is a responsive objective function for a connected network. For disconnected networks, however,  $O_{2t}$  is not good metric since it is always zero. Therefore, a ternary criterion is defined to break ties between two disconnected networks with the same value of  $O_{1t}$  as follows. First disconnected partitions (connected segments of a network) of the network are identified. Then, the midpoint be-

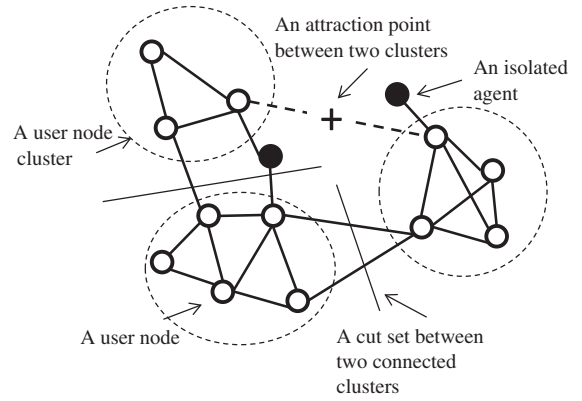


Fig. 1. A simple example for demonstrating clusters, cut sets, and an attraction point.

tween the closest nodes of two partitions is calculated for each partition pair. These midpoints are called *attraction points* since deploying an agent close to the attraction point between two disconnected partitions may connect them. Finally, the distances between agents and the attraction points are calculated and the minimum of these distances is used as the tie breaker between two disconnected networks with the same  $O_{1t}$ . This ternary criterion is expressed as follows:

$$O_{3t} = \min_{\substack{i \in AN_t \\ j \in A_t}} \left\{ \sqrt{(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2} \right\} \quad (6)$$

where  $A_t$  is the set of attraction points of the network at time  $t$ . Fig. 1 also illustrates an attraction point.

The overall objective of the proposed MANET system is first to maximize  $O_{1t}$  and then to maximize  $O_{2t}$  if a network is fully connected. Therefore, while comparing two networks in the PSO algorithm, the one with a higher  $O_{1t}$  is superior. When comparing two connected networks,  $O_{2t}$  is the dominant objective since they will have the same  $O_{1t}$  value. When comparing two disconnected networks,  $O_{1t}$  is the dominant factor. However, if two disconnected networks have the same value of  $O_{1t}$ , then the one with the higher  $O_{3t}$  is preferred. Note that  $O_{2t}$  and  $O_{3t}$  are calculated only if they are needed. There is no need to calculate  $O_{2t}$  for disconnected networks or to calculate  $O_{3t}$  for connected networks.

### 3.3. Future location prediction using kinematics

The objectives in the previous section are considered for each time step  $t$ . The proposed system is expected to maximize connectivity during a mission time between  $t_0$  and  $t_{max}$ . Therefore, predicting the future location of user nodes and deploying agent nodes according to these predictions may improve connectivity over the mission time. There are a few papers addressing the location prediction problem in MANETs. Wang and Chang [31] and Tang et al. [32] utilized a simple location and velocity based expected position to help the routing protocol. This approach does not make use of direction, change in direction nor change in velocity. The significant location learning approach

proposed by [33] predicts the next important station that the user will be in, but does not utilize velocity nor direction information. It is also only functional over the specific area that is used for learning. The neural network model proposed by Mitrovic [34] is aimed at predicting car motion. The neural network is trained using specific maneuvers on certain road conditions. However, the proposed neural network approach might be adapted for more generalized motion patterns. Although satisfactory results were achieved, the time series method of Creixell and Sezaki [35] has a time series parameter prediction problem at every time step, adding to the computational burden. In a more recent study, Huang and Zaruba [36] proposed a method that enables non-GPS equipped ad hoc nodes to estimate their approximate locations by using the information from GPS equipped nodes. This would be advantageous in situations, where GPS equipment or satellite signals were unavailable to all users.

In this paper, a location prediction method based on kinematic principles is developed for MANET users. Kinematics is a branch of mechanics which describes the motion of objects by means of only geographical coordinates, i.e. with no consideration of the forces acting on the bodies. The position of an object is described by its coordinates. The rate of change of position is defined as the velocity, and the rate of change of velocity is described as the acceleration of an object. By using the velocity and the acceleration information, it is possible to calculate how the position of an object changes by the following equations

$$\begin{aligned} x_{i(t+1)}^p &= x_{it} + v_{it}^x + \frac{1}{2} \hat{a}_{it}^x \cdot \Delta t \\ y_{i(t+1)}^p &= y_{it} + v_{it}^y + \frac{1}{2} \hat{a}_{it}^y \cdot \Delta t \end{aligned} \quad (7)$$

where  $v_{it}^x$  and  $v_{it}^y$  indicate the calculated velocity of user node  $i$  along the  $x$ -axis and  $y$ -axis, respectively, at time  $t$ , and  $\hat{a}_{it}^x$  and  $\hat{a}_{it}^y$  indicate the estimated changes in the respective velocities, i.e. acceleration, between time  $t$  and  $t+1$ . The only data needed for the position prediction of an ad hoc user is its position history from three time steps back. With position data at each time step from time  $(t-3)$ , it is possible to calculate the rate of change of acceleration, which is equivalent to the third derivative of the position, as follows:

$$\begin{aligned} v_{t-2}^x &= x_{t-2} - x_{t-3} \\ v_{t-1}^x &= x_{t-1} - x_{t-2} \\ v_t^x &= x_t - x_{t-1} \\ a_{t-1}^x &= \frac{v_t^x - v_{t-1}^x}{\Delta t} \\ a_{t-2}^x &= \frac{v_{t-1}^x - v_{t-2}^x}{\Delta t} \\ \Delta a_{t-1}^x &= \frac{a_{t-1}^x - a_{t-2}^x}{\Delta t} \\ \hat{a}_t^x &= a_{t-1}^x + \Delta a_{t-1}^x \cdot \Delta t \\ \hat{v}_{t+1}^x &= v_t^x + \hat{a}_t^x \cdot \Delta t \end{aligned}$$

The location prediction system is integrated into the mobile agent location optimizer to use past user location information.

## 4. The particle swarm optimization

Particle swarm optimization (PSO), developed by Eberhart and Kennedy ([37]), is a population based optimization tool which emulates the social behavior of species that live in the form of swarms in nature. These swarms are capable of exchanging valuable information such as food locations in the habitat. Like a genetic algorithm, PSO has a population of randomly initialized candidate solutions. Different from evolutionary algorithms, swarm particles do not mate nor mutate to create offspring. Instead, they swarm, using individual velocity vectors, over the search space while communicating with others in their neighborhood (the social factor) and using the information from their own best positions in the past (the cognition factor). The values of their positions are evaluated by the objective function. The particles “live” forever, that is, there is no creation or removal of particles during the optimization.

The positioning of agents in mobile ad hoc networks is a dynamic optimization problem with time varying problem data. A heuristic approach to dynamic optimization is expected to locate and track the changing optimum at each time step. Among approaches to dynamic optimization, population based meta-heuristic algorithms, such as evolutionary algorithms and PSOs, are well suited for dynamic optimization problems. A survey on evolutionary algorithms for problems with changing environments is given by [38]. PSO is especially well suited for solving dynamic optimization problems due to its prowess in controlling population diversity and convergence [39]. The first PSO approach to dynamic problems was reported by [40]. Since then, PSO has been successfully applied to various dynamic optimization problems [41–47]. For the application in this paper, the only change to a standard PSO is to fix the agents' locations from the previous time step's best solution into the next time step's swarm.

### 4.1. Decision variables and objective function

At any time  $t$ , the location of user nodes ( $UN_t$ ) and agent nodes ( $AN_t$ ) are known. The decision problem is to relocate the agents from their current positions at time  $t$  to their optimized locations at time  $t+H$  in order to maximize network connectivity at time  $t+H$ . In other words, the PSO algorithm searches for the best location ( $x_{i(t+H)}, y_{i(t+H)}$ ) of each agent node  $i$  at time  $t+H$ . Clearly, the distance that an agent can cover during the time span of  $H$  is limited. The relocation decision of agent  $i$  at time  $t$  is represented in the polar coordinate system as ( $r_{it}, v_{it}$ ), where  $r_{it}$  and  $v_{it}$  are the rotation angle (in radians, counter clockwise) from the  $x$ -axis and the distance that agent  $i$  travels at direction  $r_{it}$  in  $H$  time units, respectively. The Cartesian coordinates of an agent at time  $(t+H)$  can be calculated as

$$\begin{aligned} x_{i(t+H)} &= x_{it} + \cos(r_{it})v_{it} \quad \forall i \in AN_t \\ y_{i(t+H)} &= y_{it} + \sin(r_{it})v_{it} \quad \forall i \in AN_t \end{aligned} \quad (8)$$

There is an important advantage of implementing a polar coordinate system when representing the mobile agent relocations. The rotation  $r_{it}$  is bounded by  $[0, 2\pi]$  and the

distance  $v_{it}$  is bounded by  $[0, H v_{\max}]$ . This allows the PSO algorithm to move the mobile agents freely within a circle of radius  $H v_{\max}$ , thus automatically complying with the maximum distance constraints. Using the rotation and distance decision variables, the agent location optimization problem, termed ALOC, solved at time  $t$  is given as follows:

Problem ALOC( $t, H$ ):

$$\begin{aligned} \max z &= M O_{1(t+H)} + O_{2(t+H)} \\ x_{i(t+H)} &= x_{it} + \cos(r_{it}) v_{it} \quad \forall i \in AN_t \\ y_{i(t+H)} &= y_{it} + \sin(r_{it}) v_{it} \quad \forall i \in AN_t \\ r_{it} &\leq 2\pi \quad \forall i \in AN_t \\ v_{it} &\leq H v_{\max} \quad \forall i \in AN_t \\ r_{it}, v_{it} &\geq 0 \end{aligned}$$

where  $M$  is large number to make sure that  $O_{1t}$  is always weighted more than  $O_{2t}$ .  $M$  should be selected large enough so that  $O_{1t}$  is always the primary objective and  $O_{2t}$  is the secondary one. In other words,  $O_{2t}$  should not be improved at the expense of  $O_{1t}$ . In this paper  $M$  is selected to be larger than the product of the number of the user node pairs  $(NU \times (NU - 1)/2)$  and the maximum possible data rate of a link. For example, for a 10-node network with the maximum of link capacity of 54 Mbps,  $M$  is set to  $45 \times 54000$ .  $M$  can be omitted in the objective function if all user node pairs are considered in the evaluation of  $O_{2t}$  (i.e. when  $O_{2t}$  is not a surrogate) since  $O_{2t}$  becomes zero when  $O_{1t} < 1$ .  $O_{3t}$  does not appear in the objective function since it is a tie breaker criterion that is used in the PSO algorithm.

#### 4.2. Optimizing time step to time step

As described above, the PSO algorithm searches for the best location  $(x_{i(t+H)}^*, y_{i(t+H)}^*)$  of each agent node  $i$  at time  $t + H$ . However, due to the dynamic nature of the problem, at time  $t + 1$ , which is the start of a new optimization cycle, new location information of the users becomes available and the optimization problem is solved again. Based on the deployment decision at time  $t$ , the locations of the agents at time  $t + 1$  are determined as follows:

$$\begin{aligned} x_{i(t+1)} &= x_{it} + \cos(r_{it}^*) \min\{v_{\max}, v_{it}^*\} \quad \forall i \in AN_t \\ y_{i(t+1)} &= y_{it} + \sin(r_{it}^*) \min\{v_{\max}, v_{it}^*\} \quad \forall i \in AN_t \end{aligned} \quad (9)$$

Eq. (9) means that each agent node  $i$  is deployed at time  $t$  with a commitment to be at location  $(x_{i(t+H)}^*, y_{i(t+H)}^*)$  at time  $(t + H)$ ; however, at the beginning of time period

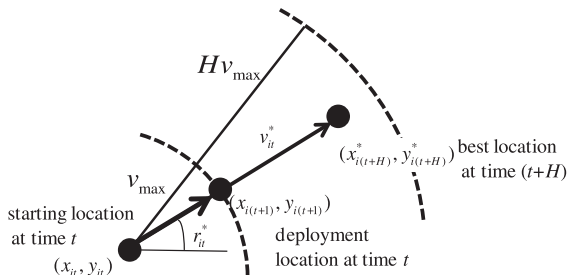


Fig. 2. Decision variables and deployment constraints of an agent.

$t + 1$ , the actual locations of the user nodes now become available therefore, the deployment decision needs to be revised accordingly. Although the optimization is performed for  $H$  time steps ahead, the decision implementation is in the next time step as shown in Fig. 2. In other words, an agent  $i$  is allowed to travel a maximum of  $v_{\max}$  distance at the direction of  $r_{it}^*$ . The overall pseudo code of the MANET optimization system is given below.

#### Procedure MANET Optimization System()

```

Start {
  Initialize  $t = t_0$  and choose  $H$ 
  Read the initial locations  $(x_{it}, y_{it})$  for  $\forall i \in N_t$ 
  Do{
    Predict  $(x_{i(t+H)}, y_{i(t+H)})$  for  $\forall i \in UN_t$ 
    Solve ALOC( $t, H$ ) to determine  $r_{it}^*$  and  $v_{it}^*$  for
     $\forall i \in AN_t$ 
    Find  $(x_{i(t+1)}, y_{i(t+1)})$  according to (9) using  $r_{it}^*$  and  $v_{it}^*$ 
    for  $\forall i \in AN_t$ 
    Deploy mobile agents to  $(x_{i(t+1)}, y_{i(t+1)})$  for  $\forall i \in AN_t$ 
    Read new actual location data  $(x_{i(t+1)}, y_{i(t+1)})$  for
     $\forall i \in UN_t$ 
    Create network  $G_{t+1}$ 
    Evaluate the actual performance of  $G_{t+1}$ 
    Set  $t = t + 1$ 
  } While (User nodes are active)
} End
    
```

#### 4.3. PSO parameters

Since the relocation decision of an agent  $i$  at a time  $t$  is represented in the polar coordinate system as  $(r_{it}, v_{it})$ , each particle  $j$  in the swarm is represented by three vectors:  $X[t, j] = \{(r_{ij}, v_{ij}) : i \in AN_t\}$ ,  $X^*[t, j] = \{(r_{ij}^*, v_{ij}^*) : i \in AN_t\}$ ,  $\Delta X[t, j] = \{(\Delta r_{ij}, \Delta v_{ij}) : i \in AN_t\}$ . At time  $t$ , the vector  $X[t, j]$  represents the current location of particle  $j$  in the solution space,  $X^*[t, j]$  represents the location of the particle's best fitness during the optimization cycle between times  $t$  and  $t + 1$ , and  $\Delta X[t, j]$  is the velocity vector that defines the direction and magnitude that the particle will travel in the solution space if not disturbed.  $\Delta X[t, j]$  is used to update  $X[t, j]$  at every iteration of the PSO as follows:

$$\begin{aligned} r_{ij} &= r_{ij} + \Delta r_{ij} \quad \forall i \in AN_t \\ v_{ij} &= v_{ij} + \Delta v_{ij} \quad \forall i \in AN_t \end{aligned} \quad (10)$$

Since the search in PSO is controlled by the modification of particle velocities, controlling the changes in the velocity is a major concern. If left unbounded, the magnitudes of particle velocities can reach quite large numbers [48]. There are two main methods to control the changes in particle velocities: (i) an inertia coefficient and (ii) a constriction coefficient. In the PSO algorithm herein, both methods are utilized to control particle velocities.

The inertia method employs a dynamic inertia ( $\omega$ ) coefficient [48]. In this paper,  $\omega$  is initially set to 1.5 and is decreased geometrically by a coefficient of 0.98 at each iteration. Although the common practice is to let the

inertia coefficient decrease monotonically, from the preliminary experimentation, it was found beneficial to occasionally reset it to its initial value. This improves the ability to escape local optima by “exciting” the particles every now and then, helping them swarm to other regions. The inertia coefficient is reset to its original value of 1.5, with a probability of 0.02, for each particle.

The constriction coefficient,  $K$ , was developed by Clerc [49]. It scales velocity updates such that a theoretical convergence is guaranteed. It has been found that  $K$  combined with velocity constraints can improve PSO performance significantly [50]. The constriction coefficient  $K$  is implemented as given below:

$$\begin{aligned}\Delta r_{ij} &= K(\omega \Delta r_{ij} + \phi_1 R_1 (r_{it}^* - r_{ij}) + \phi_2 R_2 (r_{it}^* - r_{ij})) \\ \Delta v_{ij} &= K(\omega \Delta v_{ij} + \phi_1 R_1 (v_{it}^* - v_{ij}) + \phi_2 R_2 (v_{it}^* - v_{ij}))\end{aligned}\quad (11)$$

where  $\omega$  is the inertia factor,  $\phi_1$  and  $\phi_2$  are the cognition and the social coefficients, respectively,  $R_1$  and  $R_2$  are uniform random numbers between  $[0, 1]$ , and  $r_{it}^*$  and  $v_{it}^*$  are the rotation angle and velocity of the best particle so far during the search at time  $t$ . The constriction coefficient  $K$  is calculated in each iteration as

$$\begin{aligned}\phi &= \phi_1 R_1 + \phi_2 R_2 \\ K &= \begin{cases} \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} & \phi > 4 \\ 1 & \text{otherwise} \end{cases}\end{aligned}\quad (12)$$

The swarm particle velocities are limited to the variable limits, which is a common practice in PSO. One important thing that needs to be noted is that the swarm particle velocity vector is scaled down entirely. That is, all of its elements are scaled down rather than only the ones that exceed the limit. The scale factor for a particle  $j$  is calculated using the velocity element that has the largest deviation from the maximum allowed velocity limit as follows:

$$\rho_{jt} = \max \left( \max_{i \in AN_t} \left( \frac{|\Delta r_{ij}|}{\Delta r^{\max}} \right), \max_{i \in AN_t} \left( \frac{|\Delta v_{ij}|}{\Delta v^{\max}} \right) \right)$$

where  $\Delta r^{\max}$  and  $\Delta v^{\max}$  are maximum allowed-absolute velocities for the rotation and distance decision variables, respectively. For a particle  $j$ , if  $\rho_{jt} > 1$ , then at least one of the velocities violates the velocity limits. In this case, the velocities for particle  $j$  are scaled down as follows:

$$\begin{aligned}\Delta r_{ij} &= \Delta r_{ij} / \rho_{jt} \quad i \in AN_t \\ \Delta v_{ij} &= \Delta v_{ij} / \rho_{jt} \quad i \in AN_t\end{aligned}\quad (13)$$

On the other hand, if  $\rho_{jt} \leq 1$ , no scaling is necessary. As a final PSO setting, the global neighborhood topology is used, that is, the entire swarm forms a single neighborhood. This strategy has been found to be the most efficient over a wide variety of continuous test problems as suggested by [51].

#### 4.4. Initialization and operation of the PSO

At the beginning of a time step  $t$ , the inputs to the PSO algorithm are: (i) the deployment decision (i.e.,  $r_{i(t-1)}^*$  and  $v_{i(t-1)}^*$  for  $\forall i \in AN_{(t-1)}$ ), and (ii) the estimated user node locations at time  $t+H$ . The deployment decision of an agent  $i$  at time period  $(t-1)$  becomes the starting location of the agent node at time  $t$  (i.e.,  $(x_{it}, y_{it})$  in (9)). Each particle  $j$  is initialized as described in the procedure Initialize\_Population() below. Each particle  $j$  searches new locations for the agents at time  $(t+H)$ . At the termination of the PSO algorithm, the best particle, which is evaluated based on  $G_{t+H}$ , is returned for the deployment decision of the time step.

At the beginning of each time step  $t$ , the  $(r_{ij}, v_{ij})$  and  $(\Delta r_{ij}, \Delta v_{ij})$  vectors must be initialized for each particle  $j$ . In addition, the starting location of each agent node is set according to the best particle of the previous time step. The procedure for particle initialization is given below.

---

```

Procedure Initialize_Population( $t$ ) {
  for ( $j = 1, \dots, \mu$ ) {
    set  $x_{it} = x_{i(t-1)}^*$  and  $y_{it} = y_{i(t-1)}^*$   $\forall i \in AN_t$ 
    set  $r_{ij} = U(0, 2\pi)$  for  $\forall i \in AN_t$ 
    set  $v_{ij} = U(0, H v_{\max})$  for  $\forall i \in AN_t$ 
    set  $\Delta r_{ij} = U(-2\pi, 2\pi)$ 
    set  $\Delta v_{ij} = U(-v_{\max}, v_{\max})$ 
    set  $\Delta r^{\max} = 2\pi$ 
    set  $\Delta v^{\max} = H v_{\max}$ 
  }
}

```

---

To evaluate a particle  $j$ , the locations of agents at time  $t+H$  are calculated using (8). Then, the MANET topology  $G_{t+H}$  is formed and link capacities are calculated using (3). The procedure of particle evaluation is given below.

---

```

Procedure Evaluate_Solution( $X[t, j]$ ) {
  Calculate  $(x_{i(t+H)}, y_{i(t+H)})$  for  $\forall i \in AN_t$  using decoded  $r_{ij}$ 
  and  $v_{ij}$  values as given in (8)
  Form MANET topology  $G_{t+H}$  based on the predicted
  locations of the user nodes and calculated locations
  of the agent nodes
  Using (3), calculate the link capacity  $u_{ij(t+H)}$  for each
  link  $(i, j)$  on  $G_{t+H}$ 
  Evaluate  $O_{1(t+H)}$ ,  $O_{2(t+H)}$  and  $O_{3(t+H)}$  if necessary
  If  $(X[t, j])$  is better than  $X^*[t, j]$  then set  $X^*[t, j] = X[t, j]$ 
  If necessary, update  $r_{it}^*$  and  $v_{it}^*$  for  $\forall i \in AN_t$ 
}

```

---

In dynamic optimization problems, the stopping criteria depend on the time available for the optimization algorithm to run. In this paper, the PSO algorithm is stopped after a predetermined number of iterations. This simple stopping criteria is chosen to make sure that statistical analyses in the computational experiments are performed using the same stopping criteria.

The overall PSO procedure is:

---

```

Procedure PSO( $t$ )
{
  Initialize_Population( $t$ )
  Evaluate_Population
  Set  $\omega = 1.5$ 
  While (stopping criteria is not met) {
     $\omega = \omega \times 0.98$ 
    for ( $j = 1, \dots, \mu$ ) {
      if ( $U(0, 1) < 0.02$ ) then set  $\omega = 1.5$ 
      Calculate  $\phi$  and  $K$  using (12)
      Calculate  $\Delta r_{ij}$  and  $\Delta v_{ij}$  using (11)
      Scale  $\Delta r_{ij}$  and  $\Delta v_{ij}$ 
      Update  $r_{ij}$  and  $v_{ij}$  using (10)
      Evaluate_Solution ( $X[t, j]$ )
    }
  }
  Sort population  $P(t)$ 
  return  $r_{it}^*$  and  $v_{it}^*$  for  $\forall i \in AN_t$ 
}

```

---

## 5. Computational experiments

Both dynamic and static problems were tested. The former covers many time steps of network operation and would be of the type used in practice. The latter was used to gauge comparison with a MIP approach and to evaluate scalability to larger sized problems. For static problems, the PSO algorithm stops and returns a solution after 1000 iterations and for dynamic problems the requirement for each time step is set equal to 100.

### 5.1. Dynamic test problem simulation environment

To test the performance of the PSO algorithm under dynamic scenarios, a simulation procedure was created to randomly generate the locations of user nodes. In the simulation procedure, each user node  $i$  is assigned to a random starting point  $(x_{i0}, y_{i0})$  and a random destination point  $(x_i^T, y_i^T)$  and follows a path with random perturbations to its destination as described in the next sentences. At each time step each user is assigned to a random velocity  $U(v_{\min}, v_{\max})$ . The direction of a user node is calculated for each successive time step as shown in

$$\vec{v}_{it} = \begin{cases} \alpha v_{i(t-1)} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} + (1 - \alpha) \frac{(x_i^T, y_i^T) - (x_{i(t-1)}, y_{i(t-1)})}{|(x_i^T, y_i^T) - (x_{i(t-1)}, y_{i(t-1)})|} \\ U(0, 1) < p \\ \alpha v_{i(t-1)} + (1 - \alpha) \frac{(x_i^T, y_i^T) - (x_{i(t-1)}, y_{i(t-1)})}{|(x_i^T, y_i^T) - (x_{i(t-1)}, y_{i(t-1)})|} \\ \text{otherwise} \end{cases} \quad (14)$$

where  $\theta = U(-\pi/4, \pi/4)$  is the random rotation in the direction of the motion,  $\alpha$  is a weight factor for the current motion direction, and  $p$  is the probability of a direction change. The initial directions of user nodes are randomly determined.

This simulation procedure is a version of the Random Waypoint Model [52], which is frequently used in the literature as a benchmark mobility model to evaluate various MANET management protocols. This motion scenario models the case of the users searching for their destination or making their way around forbidden areas or obstacles, which is a reasonable representation of a search/rescue or a military operation. Other mobility models could be readily used since the user motion strategy is not an input.

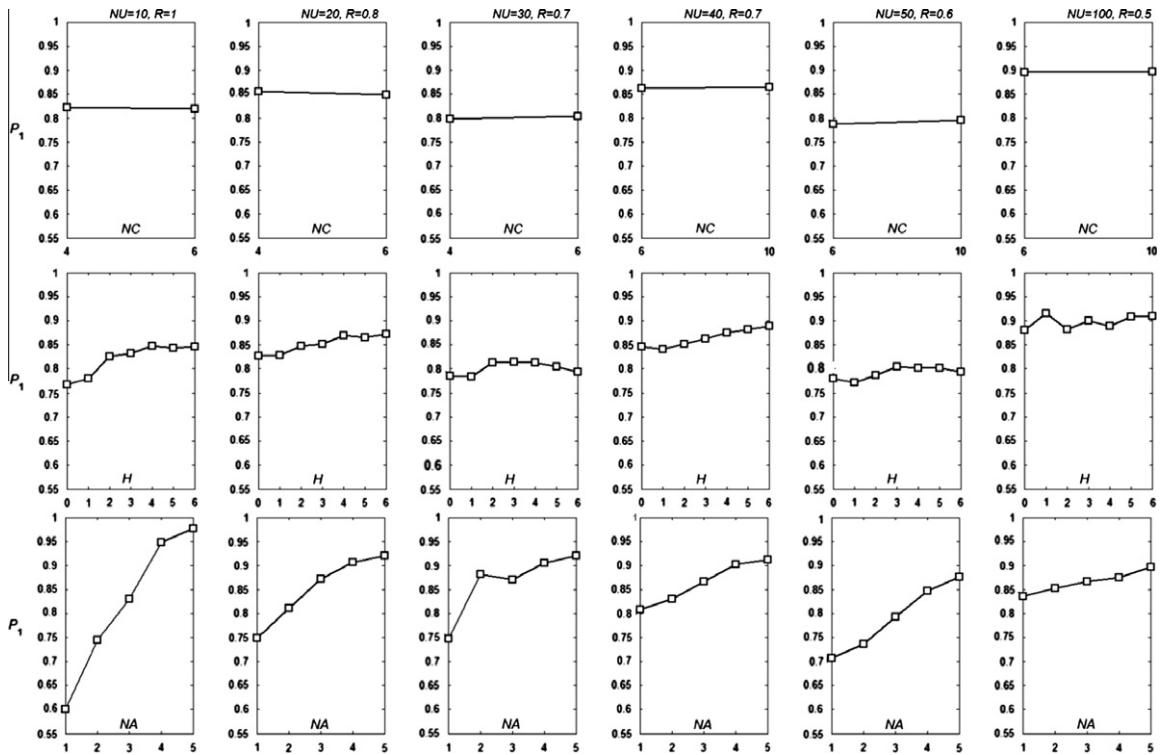
The velocity limits  $v_{\max}$  and  $v_{\min}$  for user nodes are 0.05 and 0.02, respectively. User nodes come to a stop when they reach their destinations. The velocity limits  $v_{\max}$  and  $v_{\min}$  for agents nodes are 0.06 and 0.00, respectively. The velocity values indicate Euclidean distance units traveled per one time increment. The simulation area is a two dimensional rectangular area with  $X_{\min} = 0$ ,  $X_{\max} = 5$ ,  $Y_{\min} = 0$  and  $Y_{\max} = 5$ . The simulation parameters are:  $\alpha = 0.95$  and  $p = 0.10$ .

### 5.2. Effects of number of agents, future location prediction, and clustering

Six problem groups ranging from 10-user to 100-user as given in Fig. 3 were used to test the MANET management system. Because the dimensions of the simulation are fixed, the range of user and agent nodes are set based on the number of nodes to ensure the network would be disconnected during the simulation. In each problem group, five dynamic test problems were randomly generated with a mission of 100 time steps ( $t_{\max}$ ) as described in the test problem simulation environment. In addition, each random problem was simulated five times with different random number seeds for the MANET system input parameter values of  $NA = \{1, 2, 3, 4, 5\}$ ,  $H = \{0, 1, 2, 3, 4, 5, 6\}$ , and  $NC = \{4, 6\}$  for 10-, 20-, and 30-user problems or  $NC = \{6, 10\}$  for 40-, 50-, and 100-user problems. In other words, 25 replications were performed for each combination of parameters  $NA$ ,  $H$  and  $NC$  so a total of random 1750 runs were performed for each problem group in given in Figs. 3 and 4. The population size was 50 in all runs and the PSO algorithm was run for 100 iterations during each time step of the simulation. The population size and stopping criteria are user set parameters in meta-heuristic algorithms. In practice, these parameters are experimentally determined based on the factors such as complexity of the search space, the number of decision variables, and CPU time limitations. The experimental study in this section focuses on the parameter of the proposed MANET management system. Therefore, these PSO parameters were fixed over various problems to make outcomes of the experiments independent from the optimization parameters. To evaluate the performance of the MANET management system under various system parameters, two performance metrics based on the objective functions of the problem, are defined as follows:

$$P_1 = \frac{\sum_{t=1}^{t_{\max}} (O_{1t})}{t_{\max}}$$

$$P_2 = \frac{\sum_{t=1}^{t_{\max}} \min_{i,j \in UN_t; j > i} \{U(G_t, i, j) : U(G_t, i, j) > 0\}}{t_{\max}}$$



**Fig. 3.** Main Effects Plots of performance metric  $P_1$  for random dynamic problems from 10-user to 100-user.  $P_1$  is the average user connectivity ratio (y-axis in all plots),  $NU$  = number of users,  $R$  = range,  $NA$  = number of agents,  $H$  = user location prediction time steps ahead, and  $NC$  = number of user clusters. Each column of plots belongs to a problem group stated at the top of the column. The title of the y-axis is shown only in the first column for figure clarity.

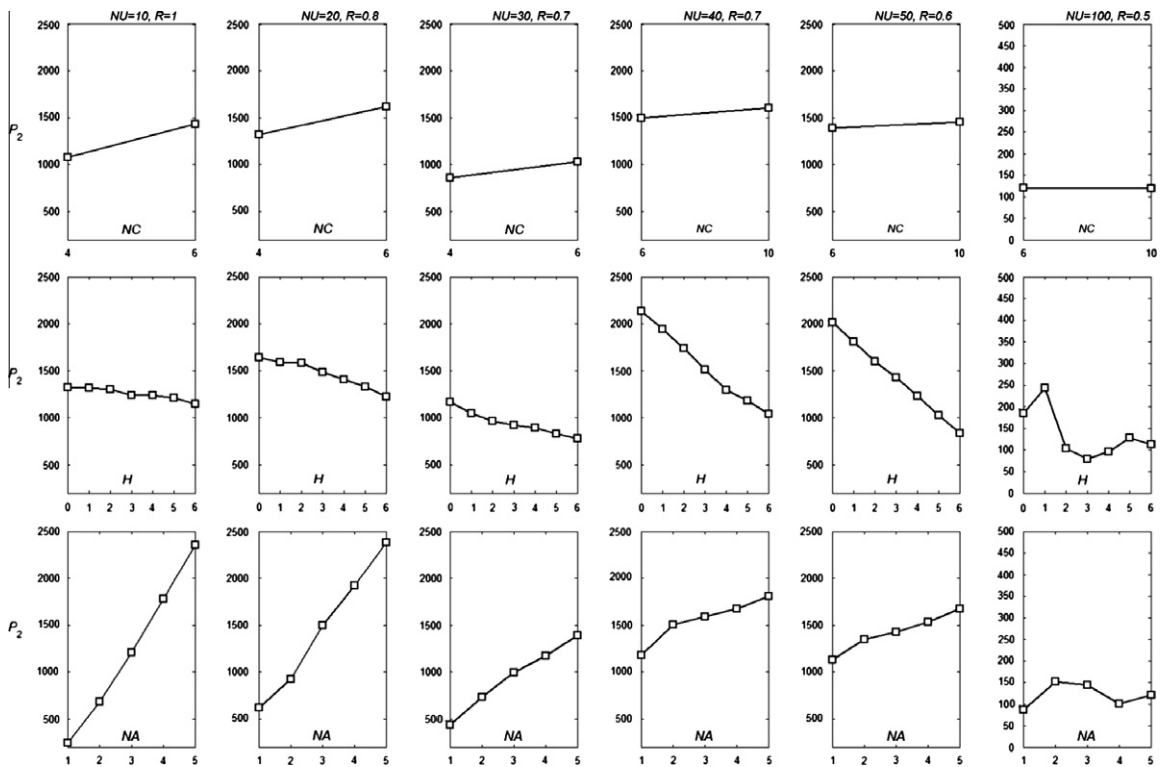
The performance metrics given in the equations above are indicators of the algorithm performance over the entire simulation time span,  $t_{\max}$ .  $P_1$  is the average user connectivity ratio and  $P_2$  is the average all-pair minimum bandwidth in bps during the simulation time span. An Analysis of Variance (ANOVA) model was used for each problem group to study the effect of  $NA$ ,  $H$  and  $NC$  on the performance of the MANET system. In statistics, an ANOVA model is used to test whether the means of several groups are all equal or not. In the ANOVA model,  $NA$ ,  $H$  and  $NC$ , and the problem instance were used as the main factors and the replications of a problem instance were used as the random factor. Fig. 3 presents the main effect plots of  $P_1$  (overall network connectivity) for the problem groups. As expected,  $P_1$  improved with an increasing number of agents ( $NA$ ) in all problems. For example in the case of 10-user problems, users were connected more than 95% of the simulation time by using four agents, as seen in Fig. 3.

As seen in the main effect plots in Fig. 3, location prediction may have a significant effect on the connectivity of the network.  $P_1$  significantly improved from  $H = 0$  (no prediction) to  $H = 3$  or 4, and then it became steady or slightly declined for higher values of  $H$ . Based on these results, predicting future locations of users is particularly important for improving overall network connectivity. However, predicting user locations into too much further into the future may have negative effects on the network connectivity because of larger prediction errors. In this experimental study,  $H = 3$  or 4 seems to be a reasonable

prediction horizon. It should be noted that the optimization problem is solved for the predicted locations of user nodes, and performance metrics  $P_1$  and  $P_2$  are evaluated using the actual locations of user nodes. Therefore, the difference between the actual and predicted user locations is expected to be the largest for  $H = 6$ .

Fig. 4 presents the main effect plots of  $P_2$  (average minimum bandwidth metric) for the problem groups. Similar to  $P_1$ , the number of agents had a significant positive effect on  $P_2$ . Although user location prediction was shown to be effective to improve overall network connectivity, it was detrimental to the average minimum bandwidth metric  $P_2$ . This result can be intuitively explained as follows. Predicting future user locations improves the overall connectivity of a network by guiding the agents to connect isolated users, which would be disconnected otherwise, to the rest of the network. The cut sets between the isolated, perimeter nodes and the others tend to have low capacities. As a result, as the network has more connected nodes, the capacity of minimum cut set tends to be smaller. Note that objective function  $O_{2t}$  and performance metric  $P_2$  only consider cut sets between the connected components of a network (i.e.  $\min_{i,j \in UN_i, j > i} \{U(G_t, i, j) : U(G_t, i, j) > 0\}$ ).  $O_{2t}$  is also the secondary objective. The PSO attempts to improve  $O_{1t}$  first; and,  $O_{2t}$  can be sacrificed in order to improve  $O_{1t}$  in long run. Therefore, increasing  $H$  has a positive effect on  $P_1$ , but a negative effect on  $P_2$ .

Fig. 5 presents the user connectivity ratio of a dynamic test problem during the course of the simulation horizon



**Fig. 4.** Main Effects Plots of performance metric  $P_2$  for random dynamic problems from 10-user to 100-user.  $P_2$  = the average all-pair minimum bandwidth in bps during the simulation time span (y-axis in all plots),  $NU$  = number of users,  $R$  = range,  $NA$  = number of agents,  $H$  = user location prediction time steps ahead, and  $NC$  = number of user clusters. Each column of plots belongs to a problem group stated at the top of the column. The title of the y-axis is shown only in the first column for figure clarity.

for  $H = \{0, 1, 6\}$  (other  $H$  values are omitted for the clarity of picture). The network lost connectivity with  $H = 6$  earlier than with  $H = 0$  because the PSO optimized the connectivity for the predicted future locations of user nodes at six time periods ahead. In other words, the connectivity was given up in the short term for a gain in the longer term. However, the PSO algorithm was able to recover or improve connectivity quickly each time that connectivity was lost for  $H = 6$ . Although user location prediction was shown to be effective to improve overall network connectivity ( $P_1$ ), prediction horizons larger than two were detrimental to the average minimum bandwidth metric ( $P_2$ ).

When using mobile agents, each agent will have associated a fixed cost and operating costs although these costs are not considered in this paper. In order to plan for the required number of agents – or resources – prior to an operation, the proposed model can be used as a tool to see the estimated network performance with different numbers of agents supporting the network.

Fig. 6 presents the network topology, the locations of the agents, and the index of user node clusters for a dynamic test problem with  $NU = 10$ ,  $NA = 1$ ,  $H = 6$  and  $NC = 4$ . Note that at the end of the simulation, the user nodes are disconnected since one user is separated from the others. In this case, there is no means to connect all user nodes with a single agent. However, the agent was deployed to maximize the connectivity of the other nodes.

Another interesting result of the experiments with dynamic problems is that the number of user node clusters ( $NC$ ) had no statistically significant effect on  $P_1$  ( $p$  values of 0.603, 0.157, 0.086, 0.355, 0.040 and 0.896 for the problems in Fig. 3, respectively) and a statistically significant, but small positive effect on  $P_2$  ( $p$ -values are close to zero in all problems in Fig. 4). Therefore, using a surrogate objective to improve the computational performance is promising. It reduces computational effort without compromising quality of the solution at each time step. Fig. 7 presents a scatter plot of  $P_2$  and the surrogate objective function for 1750 cases of the 20- and 40-user problems. As seen in the figure, there is a strong correlation between the surrogate and the actual objective function of  $P_2$  (the correlation coefficients are presented in the figure). The main advantage of using clusters of user nodes instead of individual nodes in evaluating the weakest connection of a network is computational tractability, particularly for dynamically solving large problems. Using a cluster of user nodes makes the evaluation of  $O_{t2}$  independent of the number of user nodes. Therefore, very large networks can be efficiently evaluated in short time periods. For example, the average CPU time per simulation interval is 21 s for the 100-user problems. The majority of the CPU time is due to the calculation of the maximum flows. Note that the maximum flows are calculated twice during the simulation, one for the actual surrogate  $O_{t2}$  and the other for actual

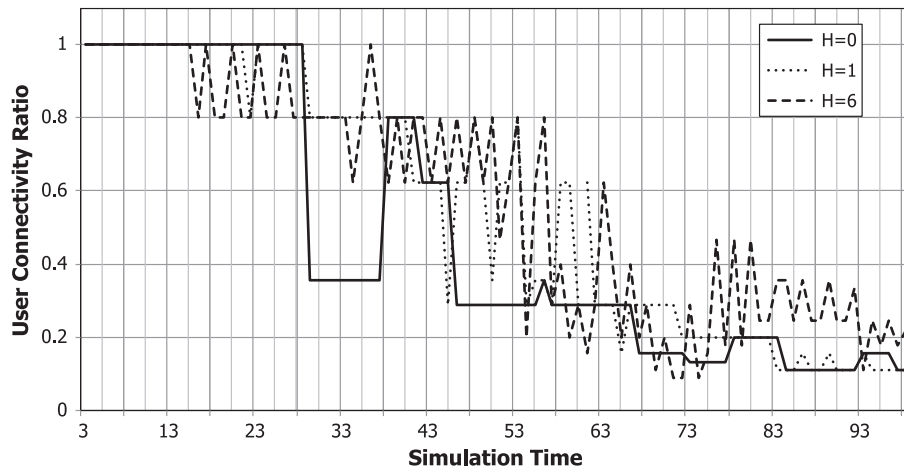


Fig. 5. User connectivity ratio of a test problem ( $NU = 10$ ,  $NA = 1$  and  $NC = 4$ ) with different user location prediction horizons.

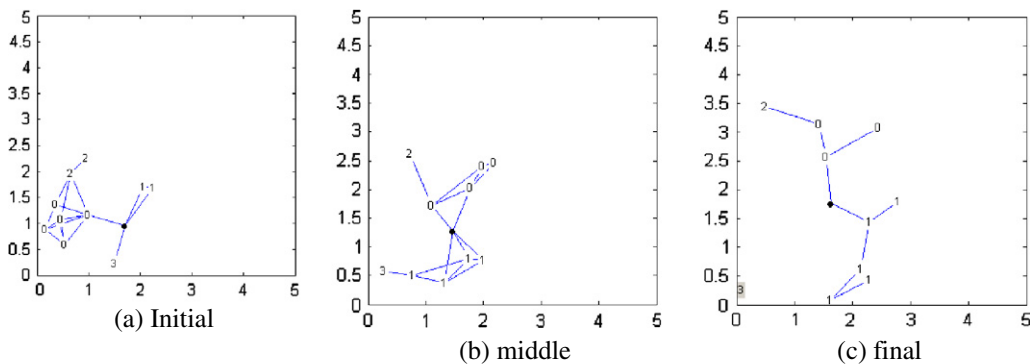


Fig. 6. Location of the users and the agent for a test problem ( $NU = 10$ ,  $NA = 1$  and  $NC = 4$ ) during the different phases of the simulation. The numbers on the link edges represent the cluster number of the users. • represents the agent node.

$O_{t2}$  to compare the performance of the surrogate. The CPU time can be significantly improved by using a more effective algorithm (e.g., the all-pairs minimum value cut algorithm [30]) for the maximum flow calculation or limiting the iterations of the PSO.

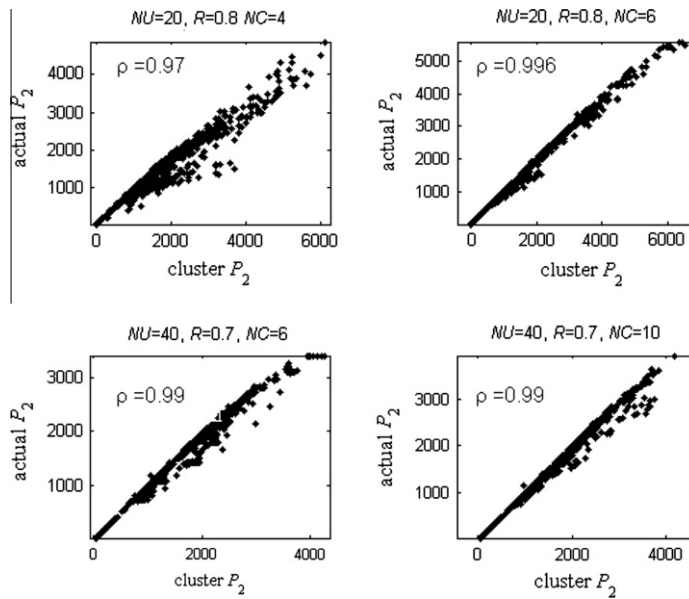
It should be noted that although several thousand runs were performed for the ANOVA, the results of this experiment should not be overly generalized.  $H$  and  $NC$  are parameters that should be selected by network managers based on empirical data and experience. Various mobility models or prediction techniques may yield different results. Nonetheless, the results of this experiment demonstrate that incorporating a user location prediction model into the proposed MANET management system may improve overall network performance in dynamic problem scenarios.

### 5.3. Comparative performance of the PSO algorithm

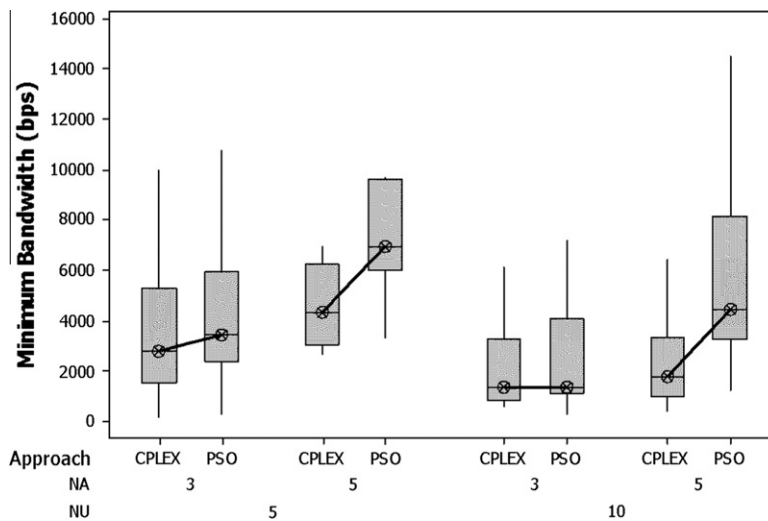
In this section, the performance of the PSO algorithm is compared with respect to a MIP formulation of the problem using static problems. Although the MIP formulation is computationally infeasible for real-life problems, the main objective of this comparison is to provide a bench-

mark for the performance of the PSO algorithm. In these problems, the locations of users were randomly generated in a way that fully connected networks were possible (i.e., the problems have feasible solutions with respect to connectivity). An approximate MIP formulation of the problem was developed using eight piecewise-linear functions for the nonlinear link distance versus capacity relationship, the Euclidean link distance and the Euclidean agent travel distances on the Cartesian coordinate system. The MIP formulation was solved by CPLEX v11 with 8 h of CPU time limit.

The MIP formulation was not able to find feasible solutions for problems with more than 10-user nodes and five agents within the allowed CPU time limit. Fig. 8 shows the comparisons on 10 random instances of 5-user and 10-user problems with three and five agents. The majority of 5- and 10-node problems with three agents were optimally solved within the CPU limit. These static problems with three agents demonstrated that the PSO performed as well as the MIP formulation. In some replications of these problems, the PSO found slightly better solutions than the MIP formulation; however the difference can be attributed to the linear approximation in the MIP formulation. As seen in figure, the PSO algorithm performed significantly



**Fig. 7.** The relationship between the surrogate and actual performance metric  $P_2$  for 20-user problems with  $NC = 4$  and  $NC = 6$  and 40-user problems with  $NC = 6$  and  $NC = 10$ . The surrogate  $P_2$  is calculated among clusters and the actual  $P_2$  is calculated over entire network.  $\rho$  is the linear correlation coefficient between the surrogate and actual  $P_2$ .



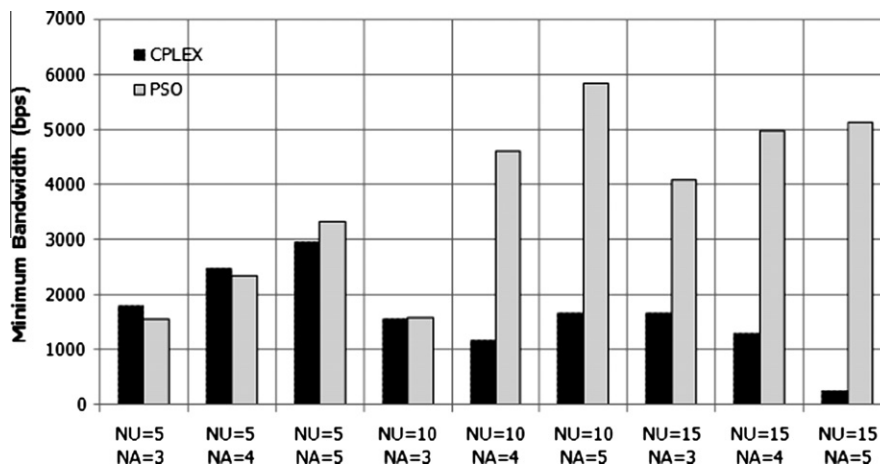
**Fig. 8.** The performance on  $P_2$  of the PSO algorithm and CPLEX on small scale static test problems in terms of minimum bandwidth ( $NA =$  number of agents and  $NU =$  number of users). Each box plot shows range, inter-quartile range, and median ( $\otimes$ ). CPLEX was run with a CPU time limit of 8 h.

better than the MIP formulation in the problems with five agents. Increasing the number of agents from three to five significantly increases the search space of the problem and these problems could not be optimally solved within the CPU time limit.

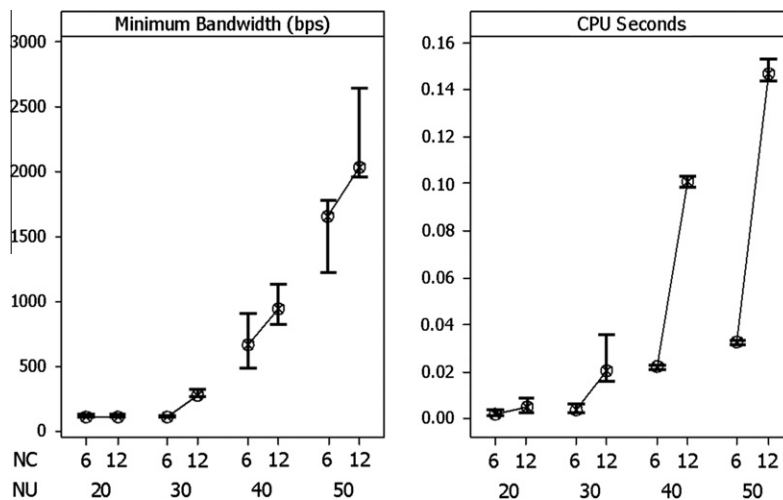
To better understand the results in Fig. 8, Fig. 9 presents a comparison of the PSO algorithm with the MIP formulation on single instances of static 5-, 10- and 15-user node problems with three, four, and five agents. In this study, CPLEX was again run for up to 8 h of CPU time and 10 random replications were performed using the PSO algorithm. The PSO results presented in the figure are the average of these 10 replications. Clearly, the MIP formulation could

not find good solutions as the problem size was even slightly increased. Note that the MIP solution for the 15-user node problem with five agents is worse than the solutions with three or four agents. This example demonstrates the challenge of the search space size of the problem. On the other hand, the PSO algorithm found superior solutions for the largest problem in this experimental study in less than 10 min of CPU time.

Finally scalability of the PSO algorithm was studied using large static problems with 20, 30, 40 and 50 user nodes with five agents. A single random problem was created for each case and solved using two different levels of the number of clusters,  $NC = 6$  and  $NC = 12$ . The outcomes



**Fig. 9.** The comparison using  $P_2$  of the PSO with the MIP formulation on single instances of static 5-, 10- and 15-user node problems with three, four, and five agents ( $NU$  = number of users and  $NA$  = number of agents).



**Fig. 10.** Summary of the results for large problems using the PSO ( $NC$  = number of clusters and  $NU$  = number of users, five agents were used for all problems) (⊗) denotes the median and bars represent the 95% confidence interval of 10 random replications. CPU seconds are given for evaluating a single solution.

of this experiment are presented in Fig. 10. The PSO algorithm was able to achieve globally connected networks in all cases; therefore, in the figure the minimum bandwidth and CPU seconds per solution are presented. While the number of clusters has no effect on the primary objective  $O_{1t}$ , maximizing the network connectivity, it did have a statistically significant effect on the minimum bandwidth (excluding the 40-user node problem). However, increasing the number of the clusters from 6 to 12 significantly increased the CPU second requirement to evaluate a solution. In dynamic cases, rigorously evaluating solutions might not be computationally feasible despite the potential gains in minimum bandwidth. In such cases, evaluating the minimum bandwidth objective based on clusters may provide significant computational benefits without sacrificing much quality of the solutions.

## 6. Conclusions

In this paper, a new model is proposed to conceptualize an autonomous topology optimization for mobile ad hoc networks using multiple mobile agents. The representation of wireless ad hoc network communications as network flows and optimization using a maximum flow model is a novel and advantageous approach. This representation is very responsive to small changes in topology when evaluating network connectivity and performance. Also, it can be used with any signal attenuation model when calculating the data flow rates.

The dynamic nature of the problem is a challenge, but it also enables the optimizer to gain additional information by leveraging the information obtained during the optimization at previous time steps. The usefulness of predicting

user locations to improve network connectivity was demonstrated. A particle swarm algorithm was developed to solve the modeled dynamic optimization problem. Computational experiments showed that the particle swarm algorithm is very promising and well suited to this problem. The PSO algorithm outperformed the MIP formulation, especially with respect to the solution time.

The proposed approach, while developed for dynamic topology optimization, easily adapts to a static scenario by increasing the agent velocity constraints. The static scenario is useful when users want to improve an existing system of sensors or communication hubs already positioned in the field, or when designing a new static system. The proposed approach could also be used for “what-if” analysis before launching an actual network in the field. The simulation would be useful to plan for the most efficient number of mobile agents to serve under a certain scenario, and to consider cost / benefit tradeoffs. As further research, the problem could be formulated as a multi-objective decision problem considering these tradeoffs.

## References

- [1] M. Forodigh et al., Wireless ad hoc networking-The art of networking without a network, *Ericsson Review* 4 (2000) 248–263.
- [2] I. Chlamtac et al., Mobile ad hoc networking: imperatives and challenges, *Ad Hoc Networks* 1 (2003) 13–64.
- [3] C.R. Dow, et al., A study of recent research trends and experimental guidelines in mobile ad-hoc network, in: Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05), 2005, pp. 72–77.
- [4] N. Aschenbruck, et al., Human mobility in MANET disaster area simulation – a realistic approach, in: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04), 2004, pp. 668–675.
- [5] D. Simplot-Ryl, I. Stojmenovic, Data communications and topology control in wireless ad hoc networks, *Ad Hoc Networks* 3 (2005) 507–508.
- [6] E. Kaplan, C. Hegarty, *Understanding GPS: Principles and Applications*, second ed., Artech House, Norwood, MA, USA, 2006.
- [7] K.H. Wang, L. Baochun, Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks, in: Proceedings of IEEE INFOCOM 2002 Conference on Computer Communications. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.37364), Piscataway, NJ, USA, 2002, pp. 1089–1098.
- [8] K.H. Wang, L. Baochun, Group mobility and partition prediction in wireless ad-hoc networks, in: Proceedings of 2002 IEEE International Conference on Communications, Piscataway, NJ, USA, 2002, pp. 1017–1021.
- [9] G. Karumanchi, et al., Information dissemination in partitionable mobile ad hoc networks, in: Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems, Los Alamitos, CA, USA, 1999, pp. 4–13.
- [10] J.A. Davis, et al., Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks, in: Proceedings of Fifth International Symposium on Wearable Computers, Los Alamitos, CA, USA, 2001, pp. 141–148.
- [11] W. Zhao, et al., A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Roppongi Hills, Tokyo, Japan, 2004.
- [12] Q. Li, D. Rus, Communication in disconnected ad hoc networks using message relay, *Journal of Parallel and Distributed Computing* 63 (2003) 75–86.
- [13] D. Goyal, J. Caffery Jr., Partitioning avoidance in mobile ad hoc networks using network survivability concepts, in: Proceedings of Seventh IEEE Symposium on Computers and Communications (ISCC'02), Taormina-Giardini Naxos, Italy, 2002, pp. 553–558.
- [14] T.-H. Kim, et al., “Improving the Topological Resilience of Mobile Ad Hoc Networks, in Proceedings of the 7th International Workshop on the Design of Reliable Communication Networks, Alexandria, Virginia, USA, 2009, pp. 191–197.
- [15] R. Montemanni, L.M. Gambardella, Swarm approach for a connectivity problem in wireless networks, in: *Swarm Intelligence Symposium*, 2005. SIS 2005. Proceedings 2005 IEEE, 2005, pp. 265–272.
- [16] C.H. Ou, et al., Connecting network partitions with location-assisted forwarding nodes in mobile ad hoc environments, in: 10th Pacific Rim International Symposium on Dependable Computing (PRDC'04), Papeete, Tahiti, French Polynesia, 2004, pp. 239–247.
- [17] K. Chadrashekar, et al., Providing Full Connectivity in Large Ad-Hoc Networks by Dynamic Placement of Aerial Platforms, University of Maryland TR, 2004–21, 2004.
- [18] H. Zhu, et al., Smart deployment/movement of unmanned air vehicle to improve connectivity in MANET, in: 2006 IEEE Wireless Communications and Networking Conference, Vegas, NV, USA, 2006, pp. 252–257.
- [19] S. Hauert et al., Ant-based swarming with positionless micro air vehicles for communication relay, *Swarm Intelligence* 2 (2008) 167–188.
- [20] C.W. Reynolds, Flocks, herds and schools: a distributed behavioural model, in: Proceedings of the Conference Held at Computer Graphics 87, Pinner, UK, 1987, pp. 71–87.
- [21] T. Balch, R. Arkin, Behavior-based formation control for multirobot teams, *IEEE Transactions on Robotics and Automation* 14 (1998) 926–939.
- [22] M. Defoort et al., Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: a distributed receding horizon approach, *Robotics and Autonomous Systems* 57 (2009) 1094–1106.
- [23] D.H. Kim et al., Decentralized control of autonomous swarm systems using artificial potential functions: analytical design guidelines, *Journal of Intelligent and Robotic Systems: Theory and Applications* 45 (2006) 369–394.
- [24] G.K. Venayagamoorthy et al., Collective robotic search using hybrid techniques: fuzzy logic and swarm intelligence inspired by nature, *Engineering Applications of Artificial Intelligence* 22 (2009) 431–441.
- [25] M. Moarref, H. Sayyaadi, Facility location optimization via multi-agent robotic systems, in: IEEE International Conference on Networking, Sensing and Control (ICNSC) 2008, 2008, pp. 287–292.
- [26] A. Rodriguez, J.A. Reggia, Extending self-organizing particle systems to problem solving, *Artificial Life* 10 (2004) 379–395.
- [27] B. McLarnon, VHF/UHF/Microwave Radio Propagation: A Primer for Digital Experimenters. 2005 November 21, 2005. <<http://www.ictp.trieste.it/~radionet/ghana1998/LINKLOSS/INDEX.HTM>>.
- [28] Netgear, ProSafe™ 802.11g Wireless Access Point, 2005, November 21, 2005. <<http://www.netgear.com/products/details/WG302.php>>.
- [29] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, *Canadian Journal of Mathematics* 8 (1956) 399–404.
- [30] R.K. Ahuja et al., *Network Flows, Theory, Algorithms, and Applications*, Prentice-Hall, Inc, Upper Saddle River, New Jersey, 1993.
- [31] N.-C. Wang, S.-W. Chang, A reliable on-demand routing protocol for mobile ad hoc networks with mobility prediction, *Computer Communications* 29 (2005) 123–135.
- [32] J. Tang, et al., Reliable routing in mobile ad hoc networks based on mobility prediction, in: Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2004, pp. 466–474.
- [33] D. Ashbrook, T. Starner, Learning significant locations and predicting user movement with GPS, in: Proceedings of the 6th International Symposium on Wearable Computers (ISWC 2002), 2002, pp. 101–108.
- [34] D. Mitrovic, Short term prediction of vehicle movements by neural networks, in: Proceedings of the 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems, 1999, pp. 187–190.
- [35] W. Creixell, K. Sezaki, Mobility prediction algorithm for mobile ad hoc network using pedestrian trajectory data, in: Proceedings of the IEEE Region 10 Conference, (TENCON 2004), 2004, pp. 668–671.
- [36] R. Huang, G.V. Zaruba, Location tracking in mobile ad hoc networks using particle filters, *Journal of Discrete Algorithms* 5 (2007) 455–470.
- [37] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Sixth International Symposium on Micro Machine and Human Science, Nagoya, 1995, pp. 39–43.

- [38] Y. Jin, J. Branke, Evolutionary optimization in uncertain environments—a survey, *IEEE Transactions on Evolutionary Computation* 9 (2005) 303–317.
- [39] W. Du, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, *Information Sciences* 178 (2008) 3096–3109.
- [40] R.C. Eberhart, Y. Shi, *Tracking and Optimizing Dynamic Systems with Particle Swarms*, Piscataway, NJ, USA, 2001, pp. 94–100.
- [41] H. Lu, W. Chen, Dynamic-objective particle swarm optimization for constrained optimization problems, *Journal of Combinatorial Optimization* 12 (2006) 409–419.
- [42] H.-W. Ge et al., A modified particle swarm optimization-based dynamic recurrent neural network for identifying and controlling nonlinear systems, *Computers and Structures* 85 (2007) 1611–1622.
- [43] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer for noisy and dynamic environments, *Genetic Programming and Evolvable Machines* 7 (2006) 329–354.
- [44] M.G.H. Omran et al., Dynamic clustering using particle swarm optimization with application in image segmentation, *Pattern Analysis and Applications* 8 (2006) 332–344.
- [45] D. Parrott, X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, *IEEE Transactions on Evolutionary Computation* 10 (2006) 440–458.
- [46] D. Weilin, L. Bin, Multi-strategy ensemble particle swarm optimization for dynamic optimization, *Information Sciences* 178 (2008) 3096–3109.
- [47] Y.-P. Chen et al., Particle swarm optimization with recombination and dynamic linkage discovery, *IEEE Transactions on Systems, Man, and Cybernetics – Part B (Cybernetics)* 37 (2007) 1460–1470.
- [48] J. Kennedy et al., *Swarm Intelligence*, Morgan Kaufmann, San Mateo, CA, 2001.
- [49] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: *Proceedings of the 1999 International Congress on Evolutionary Computation*, (ICEC 1999), Washington, DC, 1999, pp. 1951–1957.
- [50] R. C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: *Proceedings of the 2000 Congress on Evolutionary Computing*, 2000, pp. 84–88.
- [51] A. Carlisle, G. Dozier, An off-the-shelf PSO, in: *Proceedings of the 2001 Workshop on Particle Swarm Optimization*, Indianapolis, IN, 2001, pp. 1–6.
- [52] J. Broch, et al., A Performance Comparison of Multi-Hop Wireless ad hoc Network Routing Protocols, New York, NY, USA, 1998, pp. 85–97.



**Orhan Dengiz** received his bachelor's degree from Middle East Technical University (METU) in civil engineering in 2000. In August 2000, he joined Auburn University's Department of Industrial and Systems Engineering as a graduate student. He received a master's degree in 2002, and Ph.D. degree in 2007. Dengiz has worked on inter-disciplinary projects that involved materials science, computer and machine vision, and industrial engineering. In 2004, he spent one month at the Machine Vision Laboratory, University of

the West of England, Bristol, UK.

Dengiz's research involves application of computational techniques such as heuristic optimization algorithms and artificial intelligence tools in areas of manufacturing, system reliability design, automated vision systems, wireless telecommunication network design and automation problems. He published many articles in *Materials Science and Engineering*, *International Journal of Production Research*, *Manufacturing*, and *Logistics Engineering*, *Computers in Industry*, *Journal of the European Ceramic Society* and *Journal of Operational Research Society* besides many international Conference Proceedings. Publications.

After completing his doctoral studies, Dengiz returned to Ankara-TURKEY, and he has been working as a manager in a private group of companies operating in construction, energy and recycled synthetic textile production areas.



**Abdullah Konak** is an Associate Professor of Information Sciences and Technology at the Pennsylvania State University Berks. He received his degrees in Industrial Engineering, B.S. from Yildiz Technical University, Turkey, M.S. from Bradley University, and Ph.D. from the University of Pittsburgh. His current research interest is in the application of Operations Research techniques to complex problems, including such topics as telecommunication network design, network reliability analysis/optimization, facilities design, and data mining. He has published papers in *IIE Transactions*, *Operations Research Letters*, *INFORMS Journal on Computing*, *OMEGA-The International Journal of Management Science*, *IEEE Transactions on Reliability*, *International Journal of Modeling and Simulation*, *International Journal of Production Research*, *Engineering Optimization*, and *Journal of Intelligent Manufacturing*. He is a member of IIE and INFORMS.



**Alice E. Smith** is Professor and Chair of the Industrial and Systems Engineering Department at Auburn University. Previous to this position, she was on the faculty of the Department of Industrial Engineering at the University of Pittsburgh, which she joined in 1991 after industrial experience with Southwestern Bell Corporation. She has degrees in engineering and business from Rice University, Saint Louis University and Missouri University of Science and Technology.

She holds one US patent and several international patents and has authored more than 200 publications which have garnered over 1000 citations (ISI Web of Science). She won the E.L. Grant Best Paper Awards in 1999 and in 2006, and the William A.J. Golomski Best Paper Award in 2002. Several of her papers are among the most highly cited in their respective journals including the second most cited paper of *IEEE Transactions on Reliability*. She currently holds editorial positions on *INFORMS Journal on Computing*, *Computers & Operations Research*, *International Journal of General Systems* and *IEEE Transactions on Evolutionary Computation*.

She has served as a principal investigator on over \$4 million of sponsored research. Her research in analysis, modeling and optimization of complex systems has been funded by NASA, NIST, DOT/FHWA, Lockheed Martin, Adtranz (now Bombardier Transportation), the Ben Franklin Technology Center of Western Pennsylvania and NSF, from which she was awarded a CAREER grant in 1995 and an ADVANCE Leadership grant in 2001. Her industrial partners on sponsored research projects have included DaimlerChrysler Electronics, Eljer Plumbingware, Extrude Hone, Ford Motor, PPG Industries and Crucible Compaction Metals. International research collaborations have been sponsored by the federal governments of Japan, Turkey, United Kingdom, the Netherlands, Egypt, South Korea, Iraq and the US and by the Institute of International Education.

She was awarded the INFORMS WORMS Award for the Advancement of Women in OR/MS in 2009. She was named an Auburn University Philpott-WestPoint Stevens Distinguished Professor in 2001, received the Senior Research Award of the College of Engineering at Auburn University in 2001 and the University of Pittsburgh School of Engineering Board of Visitors Faculty Award in 1996.

Five of her doctoral students are in tenured positions at US universities and two of these are NSF CAREER awardees. A further three doctoral students are tenured or tenure track faculty at foreign institutions. She is a fellow of the Institute of Industrial Engineers, a senior member of the Institute of Electrical and Electronics Engineers and of the Society of Women Engineers, a member of Tau Beta Pi, the Institute for Operations Research and Management Science and the American Society for Engineering Education, and a Registered Professional Engineer in Industrial Engineering in Alabama and Pennsylvania.